S. A. Solovyev, S. Tordeux, An efficient truncated SVD of large matrices based on the low-rank approximation for inverse geophysical problems, *Sib. Èlektron. Mat. Izv.*, 2015, Volume 12, 592–609

DOI: http://dx.doi.org/10.17377/semi.2015.12.048

S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 12, стр. 592–609 (2015) УДК 519.61
DOI 10.17377/semi.2015.12.048 MSC 65F15

AN EFFICIENT TRUNCATED SVD OF LARGE MATRICES BASED ON THE LOW-RANK APPROXIMATION FOR INVERSE GEOPHYSICAL PROBLEMS

S.A. SOLOVYEV AND S. TORDEUX

Abstract. In this paper, we propose a new algorithm to compute a truncated singular value decomposition (T-SVD) of the Born matrix based on a low-rank arithmetic. This algorithm is tested in the context of acoustic media. Theoretical background to the low-rank SVD method is presented: the Born matrix of an acoustic problem can be approximated by a low-rank approximation derived thanks to a kernel independent multipole expansion. The new algorithm to compute T-SVD approximation consists of four steps, and they are described in detail. The largest singular values and their left and right singular vectors can be approximated numerically without performing any operation with the full matrix. The low-rank approximation is computed due to a dynamic panel strategy of cross approximation (CA) technique.

At the end of the paper, we present a numerical experiment to illustrate the efficiency and precision of the algorithm proposed.

Keywords: Born matrix, SVD algorithm, cross approximation (CA), low-rank approximation, high-performance computing, parallel computations.

1. Introduction

Computing the exact SVD or the T-SVD in full-arithmetic is a pretty expensive task. A standard way to compute the SVD consists of the two steps: first, the matrix is reduced to a bidiagonal form, then the SVD of the bidiagonal matrix is

Solovyev S.A., Tordeux S., An efficient Truncated SVD of large matrices based on the low-rank approximation for inverse geophysical problems.

The work is supported by RFBR (grants 14-01-31340, 14-05-31222, 14-05-00049, 14-05-93090, 15-29-06922).

Received July, 14, 2015, published September, 22, 2015.

computed. Usually, the first step is done by the Householder reflections and costs $(4/3)n^2(3m-n)$ arithmetic operations (FLOPS). The second step can be performed by an iterative refinement algorithm with stoping criteron equal to the machine precision, it takes about $O(n)$ iterations, each costs $O(n)$ FLOPS [3]. Another version of the second step is a QR algorithm for the computation of eigenvalues and takes $O(n^2)$ FLOPS [4],[10]. There are various modifications of them, using a divide-and-conquer method, preconditioned and Jacobi plane rotation methods. These ones are implemented in LAPACK routines. To compare with own developed algorithm we use the LAPACK functionality from Intel MKL. However, it is almost impossible to compute such a decomposition for large-scale problems using a robust arithmetic.

The objective of this paper is designing an efficient algorithm to compute an approximation of the T-SVD based on a low-rank arithmetic.

More precisely, in this paper we aim at looking for matrices $\overline{U}_k = \{\bar{u}_1, \dots \bar{u}_k\} \in \mathbb{C}^{I\times k}$, $\overline{V}_k = \{\bar{v}_1, \dots \bar{v}_k\} \in \mathbb{C}^{J\times k}$ and $\overline{D}_k = diag\{\overline{d}_i\}_{i=1}^k \in \mathbb{R}^{k\times k}$ which approximate U_k, V_k, D_k of the T-SVD (28) in the following sense:

I The difference between approximate singular values and exact singular values is smaller than a small parameter η_1

$$d_i - \bar{d}_i < \eta_1, \quad 1 < i \leq k \quad \text{and} \quad d_i < \delta,\ k < i \leq n \tag{1}$$

II The angles between the approximated and exact left and right singular spaces are smaller than a small parameter η_2

$$\angle(U_k, \overline{U}_k) < \eta_2 \quad \text{and} \quad \angle(V_k, \overline{V}_k) < \eta_2 \tag{2}$$

where $\angle(A, B) = \arccos(\sigma)$ with the smallest singular value σ of A^*B.

The major application of the algorithm proposed is the geophysical inverse problem. It consists in determining the physical characteristics of a propagation medium by interpreting the measured data by receivers for different sources. One of the main features of this problem is the huge number of receivers, sources and parameters of the model which is under study. Among all different techniques that exist, the Full Waveform Inversion (FWI) is one of the most costly. It consists in an interative Newton-type procedure which requires the computation of the so-called Born Matrix associated with the fullwave equation.

It is known that because of its poor conditioning this very large scale problem is difficult to solve: some (combinations) of the sources, receivers or model parameters are very important to take into account whereas some other (combinations) do not have so strong impact. The numerical method proposed in this paper needs computing the Singular Value Decomposition [14, 7] of the Born matrix to reduce the complexity of this problem by identifying the most important model parameters, sources and receivers.

This paper will is composed as follows: In Section 2, we briefly recall what is a low-rank approximation of a matrix and some algorithms to compute these approximations. Section 3 is the "core"of this paper. We describe an algorithm to compute a numerical approximation of the Truncated Singular Value decomposition based on a low-rank arithmetic. In Section 4, some numerical results are presented. These results illustrate the efficiency and accuracy of the method. There are three appendices: in Appendix A, we describe in detail the Singular Value Decomposition and the Truncated Singular Value Decomposition. Appendix B is concerned with a theoretical result about the Born acoustic matrix. We recall that the Born matrix

associated with homogeneous acoustic media can be approximated by a low-rank approximation under suitable assumptions. Finally, Appendix C briefly recalls classical results about the Chebyshev tensorial interpolation.

2. The low rank approximation of a matrix

In what follows, we will make an intensive use of the low-rank approximation.

2.1. **Definition of the ε-rank of a matrix and low-rank approximation.** Let $A \in \mathbb{C}^{I\times J}$ be a matrix with I rows and J columns with $J \leq I$. The matrix A has ε-rank k if k is the smallest integer such that there exists a matrix $A_k \in \mathbb{C}^{I\times J}$ with the rank k satisfying

$$\frac{\|A - A_k\|}{\|A\|} < \varepsilon. \tag{3}$$

When $\|\cdot\|$ is the euclidean matrix norm $\|\cdot\|_2$, the ε-rank of a matrix A is explicitly given by the number of singular values which are larger than ε. Moreover, the matrix A_k can be deduced from the T-SVD of the matrix A

$$A_k \;=\; U_k\, D_k\, V_k^* \text{ with } U_k \in \mathbb{C}^{I\times k},\; D_k \in \mathbb{R}_+^{k\times k} \text{ and } V_k \in \mathbb{C}^{J\times k}. \tag{4}$$

In reality, the computation of the T-SVD of a large-size matrix is very expensive, many authors have proposed other algorithms to compute non-optimal (in the sense of the matrix euclidean norm) low-rank approximation of a matrix A. These methods consist in looking for a matrix A_k as a product of the two matrices $B_k \in \mathbb{C}^{I\times k}$ and $C_k \in \mathbb{C}^{J\times k}$, which minimize k, satisfying

$$\frac{\|A - B_k C_k^T\|}{\|A\|} < \varepsilon. \tag{5}$$

The norm $\|A\|_2$ is given by the largest singular value of the matrix A and its computation is pretty expensive. So, we prefer to use $\|A\| = \max\limits_{i\leq I, j\leq J} |A_{i,j}|$ norm. The two common techniques to obtain this factorization are the QR factorization with pivoting of the matrix and the Cross Approximation (CA) technique which is similar to the incomplete LU factorization with pivoting. To determine ε-rank of a matrix and obtain low-rank approximation, the rank-revealing QR modification with pivoting is used (RRQR-piv) [8]. The approximate number of floating-point operations for real flavors is $(2/3)n^2(3m-n), m \geq n$. The RRQR-piv algorithm is slow (however faster than the computation of the T-SVD of a matrix A) algorithm and is almost optimal in terms of the rank k, whereas the CA algorithm is rapid, but can give rise to non-optimal k.

Lets us briefly describe the RRQR-piv algorithm and describe in detail various modifications of the CA approach.

Cross approximation. The cross approximation algorithm [11] takes the following form.

- Initialization:

$$R_0 = A \in \mathbb{C}^{I\times J}, \quad n = 0 \tag{6}$$

- While stopping-criteria $(R_n) > \varepsilon\|A\|$

Step 1. Choose a pivot $(i_\star, j_\star)$ in R_n

Step 2. Define two vector columns $b_{n+1} \in \mathbb{C}^I$ and $c_{n+1} \in \mathbb{C}^J$

$$(b_{n+1})_i = (R_n)_{i,j_\star} \quad \text{and} \quad (c_{n+1})_j = \frac{(R_n)_{i_\star,j}}{(R_n)_{i_\star,j_\star}} \tag{7}$$

Step 3. Increment n: $n = n+1$. Define the matrices $B_n \in \mathbb{C}^{I\times n}$ and $C_n \in \mathbb{C}^{J\times n}$ and

$$B_n \;=\; [b_1, b_2, \cdots, b_n] \quad \text{and} \quad C_n \;=\; [c_1, c_2, \cdots, c_n] \tag{8}$$

Step 3. Update

$$R_n \;=\; A - B_n\, C_n^T. \tag{9}$$

As a result, we obtain both matrices B_k and C_k that are involved in the low-rank approximation of A.

Like for the incomplete LU factorization algorithm, we have the following options for this algorithm:

I Total pivoting: The stopping criterion is the following:

$$\text{stopping-criterion}(R_n) = \max_{i\le I, j\le J}(|(R_n)_{i,j}|) \tag{10}$$

At each iteration, the pivot is chosen by maximizing $|(R_n)_{i,j}|$ over whole the matrix

$$|(R_n)_{i_\star,j_\star}| = \max_{i\le I,\; j\le J} |(R_n)_{i,j}| \tag{11}$$

II Dynamic panel strategy: The algorithm is more complex to describe. This corresponds to a partial pivoting. First, as for the total pivoting we define $i_\square$ and $j_\square$ such that

$$|(R_n)_{i_\square,j_\square}| = \max_{i\le I,\; j\le J} |(R_n)_{i,j}| \tag{12}$$

We then define a panel $J_\square \subset [1, J]$ of width $2K+1 \in [1, J]$:

$$J_\square = \begin{cases} [1, 2K+1] & \text{if } j_\square \le K, \\ [j_\square - K, j_\square + K] & \text{if } K < j_\square \le J-K, \\ [J-2K, J] & \text{if } j_\square > J-K. \end{cases} \tag{13}$$

As long as a maximum of $|R_n|$ over this panel is larger than ε

$$\max_{i\le I,\; j\in J_\square} |(R_n)_{i,j}| > \varepsilon, \tag{14}$$

the pivot $(i_\star, j_\star)$ will be chosen into this panel of columns

$$|(R_n)_{i_\star,j_\star}| = \max_{i\le I,\; j\in J_\square} |(R_n)_{i,j}|. \tag{15}$$

When (14) is not fulfilled anymore, another panel is considered in the same way until

$$\max_{i\le I,\; j\le J} |(R_n)_{i,j}| < \varepsilon. \tag{16}$$

III Cross pivoting: The pivot is chosen in the following way: Pick by hazard a non-zero column $(R_n)_{\cdot,j_\triangle}$ of R_n, with $j_\triangle \in J$. Define the integer $i_\star \in I$ by looking for a maximum of $|R_n|$ in this column

$$|(R_n)_{i_\star,j_\triangle}| \;=\; \max_{i\in I} |(R_n)_{i_\star,j_\triangle}| \tag{17}$$

Define the integer $j_\star \in J$ by looking for a maximum of $|R_n|$ in this row

$$|(R_n)_{i_\star,j_\star}| = \max_{j\in J} |(R_n)_{i_\star,j}| \tag{18}$$

The stopping criterion is then the following:

$$|(R_n)_{i_\star,j_\star}| > \varepsilon \tag{19}$$

Remark 1. *The search for a maximum and the update of the matrix R_n is the performance of the bottle-neck of the CA algorithm. The total pivoting strategy is much slower than the two other strategies because of*

- *the search for the maximum is made over the full matrix.*
- *at each iteration the matrix should be fully updated*

The dynamical panel strategy is more efficient since

- the search for the maximum is made over a small subset of the full matrix.
- only the panel of the matrix needs to be updated at each iteration

Since the panel is formed of a group of columns, it is important to optimize the access to memory in storing the matrix in the RAM column-by-column. If a matrix is stored rows-by-rows, the panel should be constructed of a group of rows.

The cross partial pivoting strategy is also very efficient since

- the search for the maximum is made over a cross which is a small subset of the full matrix.
- the matrix R_n does not need not to be updated but only needs to be evaluated for a small number of indices at each iteration.

3. DESCRIPTION OF THE ALGORITHM

The algorithm can be decomposed into four steps.

The **first step** consists in decomposing vertically the matrix A into blocks A_i,

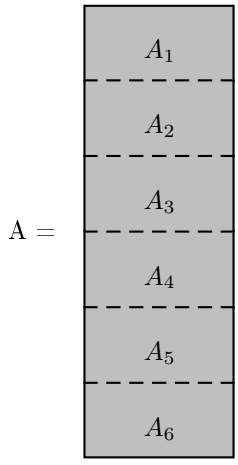

FIG. 1. Decomposition of the matrix A by blocks

(Figure 1), and in performing a low-rank approximation of each block $A_i \in \mathbb{C}^{m_i \times J}$, (Figure 2).

$$A_i \simeq B_i\, C_i^T \qquad \text{with } B_i \in \mathbb{C}^{m_i \times k_i} \text{ and } C_i \in \mathbb{C}^{J \times k_i}$$

To compute a low-rank approximation of blocks we have the three options:

i) T-SVD in the full arithmetic;

ii) RRQR-piv algorithm;

iii) Cross Approximation (CA) technique.

Remark 2. *The algorithm will only be efficient if the integer k_i is less than J. In practice, this number is small.*

Remark 3. *The accuracy of the low-rank approximation of the matrix A is characterized by a small parameter ε and by the stopping criterion. For SVD-compression, QR-piv and CA algorithm, it takes the form*

$$\begin{cases} \|A_i - B_i C_i^T\|_2 \le \varepsilon \text{ for T-SVD and RRQR algorithm,} \\ \\ \|A_i - B_i C_i^T\|_\infty \le \varepsilon \text{ for CA algorithm,} \end{cases} \tag{20}$$

with $\|\cdot\|_2$ the euclidean matrix norm and $\|A\|_\infty = \max\limits_{i \le I, j \le J} |A_{i,j}|$.

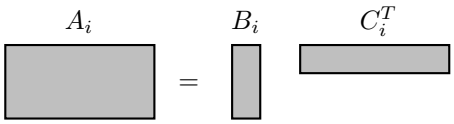

FIG. 2. Low-rank approximation of A

The result of the first step is depicted in Figure 3. In this picture and in the next ones, the "plotted"parts of matrices mean dense non-zero blocks. The "white"blocks mean zero fill-in.

At the **second step**, we orthogonalize the matrices B and C. More precisely, we perform a QR decomposition of the matrices B_i and C

$$B_i = \widetilde{B}_i\, R_i \quad \textit{and} \quad C^T = L\, \widetilde{C}^T,$$

with $\widetilde{B}_i \in \mathbb{R}^{m_i \times k_i}$, $\widetilde{C} \in \mathbb{R}^{J \times k}$ being orthogonal, $R_i \in \mathbb{R}^{k_i \times k_i}$ — the upper triangular and $L \in \mathbb{R}^{k \times k}$ — the lower triangular, where $k = \sum k_i$. The matrices $\widetilde{B}_i$ and R_i are collected into the orthogonal matrix $\widetilde{B}$ and into the upper triangular matrix R, (see Figure 4). This result in that the Low-rank approximation of A should be

$$A \simeq \widetilde{B}\, (R\, L)\, \widetilde{C}^T, \tag{21}$$

with $\widetilde{B} \in \mathbb{R}^{I \times k}$, $\widetilde{C} \in \mathbb{R}^{J \times k}$ and $RL \in \mathbb{R}^{k \times k}$ being full.

At the **third step**, a robust T-SVD with the accuracy δ of the product RL is performed

$$RL = U_{RL} D_{RL} V_{RL}^* \qquad \text{with} \quad \|U_{RL} D_{RL} V_{RL}^* - RL\| < \delta. \tag{22}$$

The result of the third step is presented in Figure 5.

Remark 4. *When the matrix RL is much smaller than the initial matrix A (this is a wide-spread in the practice case), the computation of the T-SVD in the full arithmetic of the product RL is less expensive than the computation of the T-SVD of the matrix A.*

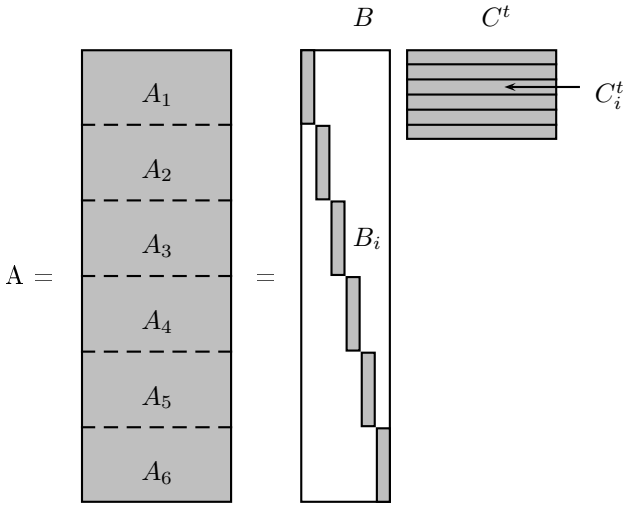

FIG. 3. Low-rank approximation of the matrix A

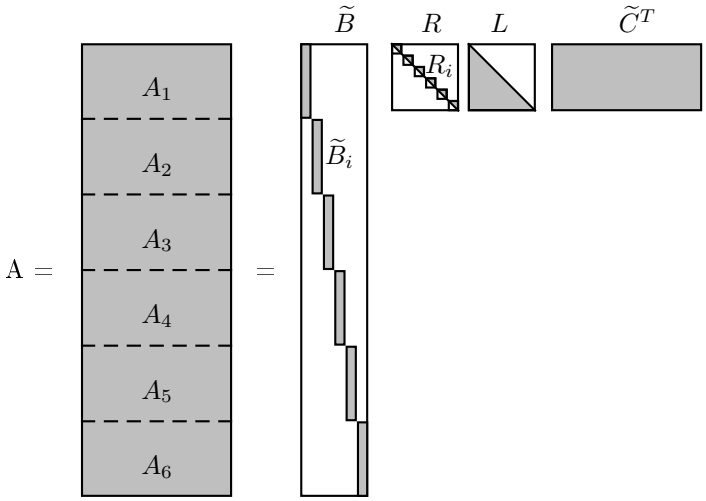

FIG. 4. The result of the second step

At the **fourth step**, we construct the final matrices by computing the products $\overline{U} = \widetilde{B}U_{RL}$, $\overline{V}^* = V_{RL}^*\tilde{C}^T$ and $\overline{D} = D_{RL}$. As a result, the matrices $\overline{U}$ and $\overline{V}$ have orthogonal columns.

Our statement is that the decomposition $\overline{U}\,\overline{D}\,\overline{V}^*$ approximates the exact T-SVD of A in the sense of (1) and (2).

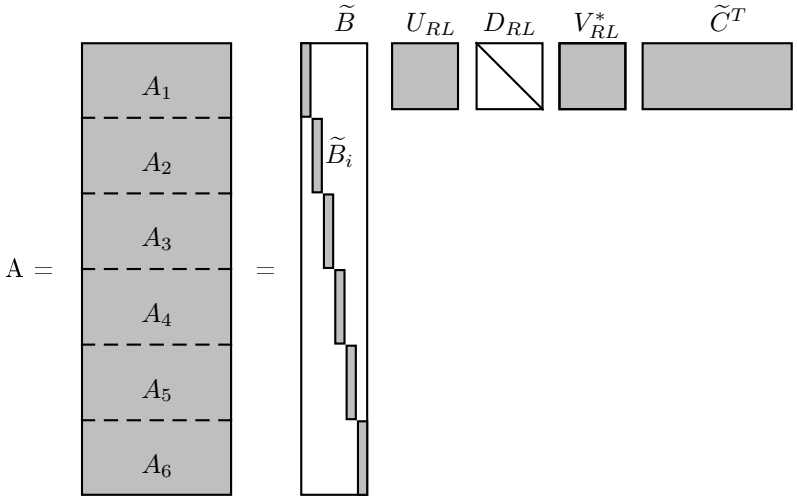

FIG. 5. The result of the third step

4. NUMERICAL EXPERIMENTS

The numerical experiments are aimed at demonstrating the performance of our algorithm in terms of computation time and precision.

We have not developed our own linear algebra library but have used the LAPACK and BLAS functions of Intel MKL. The performance was measured on Intel Core i7-3770K CPU 3.5 GHz, (Ivy Bridge). We have avoided the impact of OMP parallelization of all MKL functions by switching off threading by setting OMP_NUM_THREADS=1.

The Born matrix A associated with a 2D elasticity vertically inhomogeneous (layered) isotropic medium, one source, 1450 receivers and 10 different frequencies, has been considered. The target domain, containing 120x20 points, is a part of a huge real model which we would like to image. Details of this model are described in [13]. The full matrix A has 29,000 lines and 7,200 columns. At the preliminary step, the matrix A is separated into $p = 10$ blocks.

4.1. Computational time.

In the first test, the accuracy ε of the low-rank approximation is 10^{-6} and the threshold δ of the cropped exact SVD $U_k D_k V_k^*$ and of the cropped low-rank SVD $\overline{U}\ \overline{D}\ \overline{V}^*$ is 10^{-6}. Different options for the first step are tested: the computational time for the SVD, the QR and the CA compression methods are computed. These computational times should be compared to 970 s, i.e. the time of a robust SVD by the *gesvd* computational optimized routine of Intel MKL.

The performance results show the advantage of using partial pivoting in the CA over the cross pivoting and classical total pivoting. Additionally, the performance results confirm the high acceleration of the CA approach in comparison with the SVD and the QR-piv techniques.

TABLE 1. Comparison of performance algorithms of Low-rank SVD-based on SVD/QR/CA approaches

Steps, description \ Approaches	SVD	RRQR	CA Total pivot.	CA Cross pivot.	CA Partial pivot.
1-st, Get SVD/QR/CA of all A_i	$605s$	$255s$	$90s$	$44s$	$18s$
2-nd, Make QR of B and T-QR of C:	$18s$	$20s$	$19s$	$20s$	$25s$
3-rd, Perform SVD of RL:	$16s$	$13s$	$13s$	$13s$	$15s$
4-th, Gathering $\overline{U}$,$\overline{D}$,$\overline{V}$	$7s$	$6s$	$6s$	$6s$	$6s$
Total time	$651s$	$294s$	$130s$	$84s$	$66s$

Finally, we point out that the first step of the method proposed is easily parallelizable: since the compression of all blocks can be done by different processors as opposed to the last three steps.

4.2. **Precision.** In the second test our attention will be focussed on two different error indicators. Only the results for the most efficient case (Cross Approximation with dynamic panel partial pivoting) will be presented.

First, we are interested in the error resulted from approximate singular values. Denoting by I_ε the ε-rank of the matrix A, the following quantities

$$\begin{cases} \max\limits_{i \le I_\varepsilon} \dfrac{|d_i - \overline{d_i}|}{|d_1|} & \text{(absolute error)} \\ \max\limits_{i \le I_\varepsilon} \dfrac{|d_i - \overline{d_i}|}{|d_i|} & \text{(relative error)} \end{cases} \tag{23}$$

have been plotted with respect to ε, the parameter relating to the accuracy of the Cross Approximation parameter and with respect to δ, i.e. the final threshold parameter. The results are reported in Figure 6 for the absolute error and in Figure 7 — for the relative error. It seems that the absolute error does mostly depend on ε. On contrary, the relative error is sufficiently related to the ratio ε/δ. Second, we quantify the quality of the approximation of the left singular space. The results for the right singular values are quite similar and are not presented. We compute the angle (in degrees) between the subspace generated by the columns of $\overline{U}$ and the subspace generated by the columns of U_k

$$\angle(\overline{U}, U_k) \;=\; \arccos(\sigma)\frac{180}{\pi} \tag{24}$$

with the smallest singular value σ of the matrix $\overline{U}^* U_k$.

Numerical measurements (Figure 8) show that the angle $\alpha = \angle(\overline{U}, U_k)$, for any threshold δ, can be decreased via improving the accuracy ε of the low-rank approximation. The numerical results reveal that the error depends mostly on ε/δ like the relative error for the singular values (Figure 7).

4.3. **Evolution of ε-rank.** In the last test we investigate the ranks of truncated matrices.They are presented in Tables 2 and 3 for different low-rank approximations. The SVD approach is the most optimal in terms of ε-rank, whereas the CA technique is the worst (the rank after the first step in the Tables). For $\varepsilon = 10^{-6}$ and $\delta = 10^{-6}$, the final ranks associated with different compressions are slightly different. For

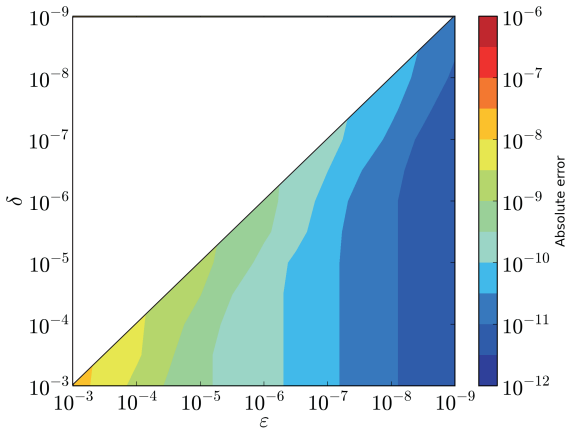

FIG. 6. The CA partial pivoting: Dependence of the singular value absolute error on accuracy ε and threshold δ.

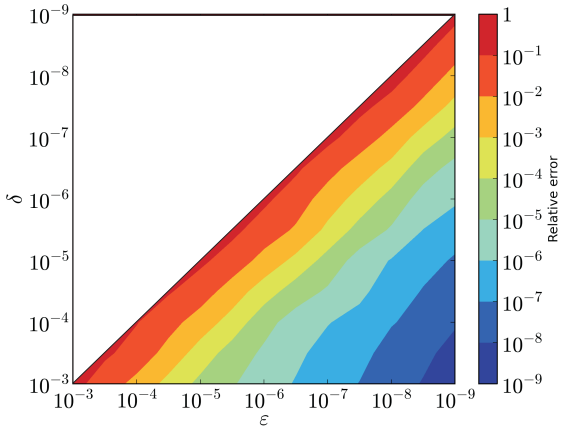

FIG. 7. The CA partial pivoting: Dependence of the singular value relative error.

$\varepsilon = 10^{-9}$ and $\delta = 10^{-6}$, the final ranks are equal. This will be the case when the compression ε is much smaller than δ. The final rank of a low-rank SVD should be compared to a rank obtained by truncated full arithmetic SVD, i.e. 1984. In the context of seismic imaging, a small misfit of the rank will not have, in our opinion, a large impact on the solution of the inverse problem.

5. CONCLUSION

We have presented an algorithm to compute the truncated SVD of the Born matrix. This method is based on a low-rank arithmetic and the CA technique. To perform the low-rank approximation, we have proposed a dynamic panel CA algorithm. This approach is similar to the panel local pivoting LU decomposition

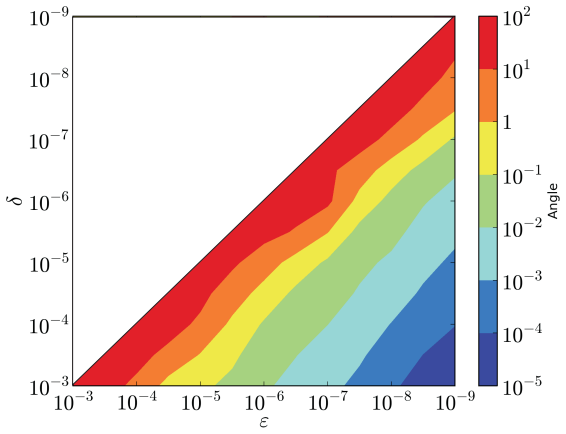

FIG. 8. The CA partial pivoting: Dependence of the angle between subspaces spanned on singular vectors of the cropped exact SVD and Low-rank SVD.

TABLE 2. The rank of matrices at intermediate steps, $\varepsilon = \delta = 10^{-6}$

Matrix rank \ Approaches	SVD	RRQR	CA Total pivot.	CA Cross pivot.	CA Partial pivot.
After the 1-st step	2366	2424	2444	2454	2795
After the 2-nd step	2041	2041	2022	1961	2013
After the 3-th step	1963	1963	1953	1936	1950

TABLE 3. The rank of matrices at intermediate steps, $\varepsilon = 10^{-9}$, $\delta = 10^{-6}$

Matrix rank \ Approaches	SVD	RRQR	CA Total pivot.	CA Cross pivot.	CA Partial pivot.
After the 1-st step	2963	3030	3052	3052	3565
After the 2-nd step	2619	2619	2586	2541	2581
After the 3-th step	1964	1964	1964	1964	1964

technique [12]. The algorithm proposed is an alternative to a very popular and efficient randomized SVD approach proposed by Rokhlin [6]. The main advantages are: (i) the ε-rank of a matrix has not to be known in advance, (ii) the computation of a reduced matrix is less expensive (this has been confirmed by preliminary numerical tests which are not included in this paper).

For a representative configuration we have compared the results generated by the proposed truncated SVD algorithm to the results obtained by an exact SVD. We have observed that the method is accurate and the acceleration of the computation has increased by the factor 10 on one-thread systems. The algorithm has a good

opportunity for parallelization both on shared memory systems (using OMP parallelization) and on distributed ones (MPI parallelization).

Appendix A. The Singular Value Decomposition and the Truncated Singular Value Decomposition

The SVD of a matrix $A \in \mathbb{C}^{I\times J}$ with $J \leq I$ is a factorization of the form

$$A = UDV^*, \tag{25}$$

where the matrices U and V contain the left and right singular vectors u_i and v_i; the matrix D is diagonal and contains the singular values d_i

$$\begin{aligned} U &= \{u_1, \cdots, u_J\} \in \mathbb{C}^{I\times J} \qquad \text{with } U^*U = I \in \mathbb{C}^{J\times J} \\ V &= \{v_1, \cdots, v_J\} \in \mathbb{C}^{J\times J}, \quad \text{with } V^*V = I \in \mathbb{C}^{J\times J} \\ D &= diag\{d_i\}_{i=1}^{J} \in \mathbb{R}_+^{J\times J}. \end{aligned} \tag{26}$$

The singular values d_i are all positive and ordered

$$d_1 \geq d_2 \geq \cdots \geq d_J \geq 0. \tag{27}$$

The T-SVD of the matrix A is obtained from the SVD by removing the singular values $d_{k+1}, d_{k+2}, \ldots$ which are lower than a small parameter δ

$$A_k = U_k D_k V_k^*, \tag{28}$$

$$\begin{aligned} U_k &= \{u_1, \cdots, u_k\} \in \mathbb{C}^{I\times k} \qquad \text{with } U_k^*U_k = I \in \mathbb{C}^{k\times k} \\ V_k &= \{v_1, \cdots, v_k\} \in \mathbb{C}^{J\times k} \qquad \text{with } V_k^*V_k = I \in \mathbb{C}^{k\times k} \\ D_k &= diag\{d_i\}_{i=1}^{k} \in \mathbb{C}^{k\times k}. \end{aligned} \tag{29}$$

The matrix A_k is an approximation of the matrix A in the following sense

$$\|A - A_k\|_2 \leq \delta \tag{30}$$

with the Euclidean matrix norm $\|\cdot\|_2$:

$$\|A\|_2 \;=\; \sup_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \tag{31}$$

Appendix B. Some theoretical results involving the acoustic Born matrix

Let us show on a simple example that the Born matrix of an acoustic problem can be approximated by a low-rank approximation derived thanks to a kernel independent multipole expansion.

The model parameters. The considered propagation domain consists of unbounded three-dimensional (3D) acoustic media governed by the Helmholtz equation with varying physical characteristics $\mu(\mathbf{y})$ (the square of the wave-number).

The model is parametrized on a regular grid with the spatial step $\delta_y \in \mathbb{R}_+$ composed of J cells $K_j \subset \mathbb{R}^3$ (Figure 9),

$$K_j \;=\; [j_1\delta_y, (j_1+1)\delta_y] \times [j_2\delta_y, (j_2+1)\delta_y] \times [j_3\delta_y, (j_3+1)\delta_y] \tag{32}$$

with the integer $j \in [1, J]$ related to the integers $j_1 \in [0, J_1 - 1]$, $j_2 \in [0, J_2 - 1]$ and $j_3 \in [0, J_3 - 1]$ by the relation

$$j \;=\; j_3\, J_2\, J_1 \;+\; j_2\, J_1 \;+\; j_1 \;+\; 1 \quad \text{ and } \quad J = J_1\, J_2\, J_3. \tag{33}$$

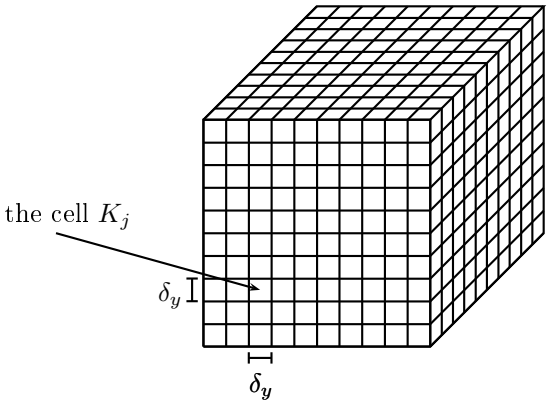

FIG. 9. Discretization of the model parameter via a 3D grid

The function μ is chosen to be constant outside the regular grid and piecewise constant on the regular grid with the value $\mu_j \in \mathbb{R}_+$ on K_j

$$\mu(\mathbf{y}) = \mu_j \text{ if } \mathbf{y} \in K_j \quad \text{and} \quad \mu(\mathbf{y}) = \mu_0 \text{ else.} \tag{34}$$

The model parameters μ_j, $1 \leq j \leq J$, are collected into a vector $\boldsymbol{\mu} \in \mathbb{R}^J$.

The data are obtained thanks to I_1 experiments, each corresponding to a source located at a point $\mathbf{x}^1_{i_1} \in \mathbb{R}^3$, with $0 \leq i_1 \leq I_1 - 1$. For every experiment, I_2 measurements are realized by receivers located at a point $\mathbf{x}^2_{i_2} \in \mathbb{R}^3$, with $0 \leq i_2 \leq I_2 - 1$. This gives rise to a set of data composed of $I = I_1\, I_2$ measurements. To organize these data, every receiver–source couple is indexed by an integer $i \in [1, I]$ given by

$$i = i_1\, I_1 \, + \, i_2 \, + 1.$$

The full wave inverse problem takes the form:

> Given $\mathbf{f} \in \mathbb{C}^I$, find $\boldsymbol{\mu} \in \mathbb{R}^J_+$ such that
>
> $$F_i(\boldsymbol{\mu}) \; = \; f_i \quad \text{for } 1 \leq i \leq I \tag{35}$$
>
> with $F_i(\boldsymbol{\mu}) = u_{i_1}(\boldsymbol{\mu}; \mathbf{x}_{i_2})$ where $\mathbf{x} \mapsto u_{i_1}(\boldsymbol{\mu}, \mathbf{x})$ is defined over all $\mathbb{R}^3$ as the outgoing solution of the direct acoustic problem for the i_1^{th} source :
>
> $$\triangle u_{i_1}(\boldsymbol{\mu}; \mathbf{x}) \; + \; \mu(\mathbf{x})\, u_{i_1}(\boldsymbol{\mu}; \mathbf{x}) \; = \; -\delta_{\mathbf{x}^1_{i_1}} \tag{36}$$
>
> with $\delta_{\mathbf{x}^1_{i_1}}$ the Dirac delta function located at the point $\mathbf{x}^1_{i_1}$.

Most of the algorithms that have been proposed in the literature are of the Newton type [9, 1]. They require the computation of the Born Matrix $A \in \mathbb{C}^{I \times J}$ which contains the partial derivative with respect to μ_j of the nonlinear form F_i.

$$A_{i,j} = \frac{\partial F_i}{\partial \mu_j}(\boldsymbol{\mu}) \quad \text{for } 1 \leq i \leq I \text{ and } 1 \leq j \leq J. \tag{37}$$

The matrix A can be expressed as

$$A_{i,j} \; = \; u^j_{i_1}(\mathbf{x}^2_{i_2}) \tag{38}$$

with respect to the function $u_{i_1}^j$ which is the partial derivative of u_{i_1} with respect to μ_j

$$u_{i_1}^j(\boldsymbol{\mu};\mathbf{x}) \;=\; \frac{\partial u_{i_1}}{\partial \mu_j}(\boldsymbol{\mu};\mathbf{x}) \;=\; \lim_{h\to 0} \frac{u_{i_1}(\boldsymbol{\mu}+he_j;\mathbf{x}) - u_{i_1}(\boldsymbol{\mu};\mathbf{x})}{h}. \tag{39}$$

Deriving (36) with respect to μ_j, we obtain that the function $u_{i_1}^j$ is the unique outgoing solution of

$$\triangle u_{i_1}^j(\boldsymbol{\mu};\mathbf{x}) \;+\; \mu(\mathbf{x})\, u_{i_1}^j(\boldsymbol{\mu};\mathbf{x}) \;=\; -\mathbf{1}_{K_j}(\mathbf{x}) u_{i_1}(\boldsymbol{\mu};\mathbf{x}) \tag{40}$$

with the characteristic function $\mathbf{1}_{K_j}$ associated to K_j

$$\mathbf{1}_{K_j}(\mathbf{x}) \;=\; 1 \text{ for } x \in K_j \text{ and } \mathbf{1}_{K_j}(\mathbf{x}) \;=\; 0 \text{ for } x \notin K_j. \tag{41}$$

This Born matrix can then be related to the Green function associated with the acoustic media. This function, which depends on $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{y} \in \mathbb{R}^3$, is symmetric $G(\boldsymbol{\mu};\mathbf{x},\mathbf{y})$ and is defined for every $\mathbf{y} \in \mathbb{R}^3$ as $G(\boldsymbol{\mu};\mathbf{x},\mathbf{y}) = G_{\boldsymbol{\mu};\mathbf{y}}(\mathbf{x})$ with $G_{\boldsymbol{\mu};\mathbf{y}}$ the outgoing solution of

$$\Delta G_{\boldsymbol{\mu};\mathbf{y}}(\mathbf{x}) \;+\; \mu(\mathbf{x})\, G_{\boldsymbol{\mu};\mathbf{y}}(\mathbf{x}) \;=\; -\delta_{\mathbf{y}}(\mathbf{x}) \quad \text{ on } \mathbb{R}^3 \tag{42}$$

with the Dirac generalized function $\delta_{\mathbf{y}}$ at $\mathbf{x} = \mathbf{y}$. It follows that the function u_{i_1}, which solves (36), is explicitly given by

$$u_{i_1}(\boldsymbol{\mu};\mathbf{x}) = G(\boldsymbol{\mu},\mathbf{x},\mathbf{x}_{i_1}^1). \tag{43}$$

Its partial derivative $u_{i_1}^j$, which solves (40) with respect to $\boldsymbol{\mu}$ is given by the representation formula

$$u_{i_1}^j(\mathbf{x}) \;=\; \int_{\mathbb{R}^3} G(\boldsymbol{\mu};\mathbf{x},\mathbf{y})\mathbf{1}_{K_j}(\mathbf{x}) u_{i_1}(\boldsymbol{\mu};\mathbf{y}) d\mathbf{y}. \tag{44}$$

Taking into account (43), we obtain

$$u_{i_1}^j(\mathbf{x}) \;=\; \int_{K_j} G(\boldsymbol{\mu};\mathbf{x},\mathbf{y}) G(\boldsymbol{\mu};\mathbf{y},\mathbf{x}_{i_1}^1) d\mathbf{y}. \tag{45}$$

This leads to the following simple formula for the Born matrix

$$A_{i,j} \;=\; u_{i_1}^j(\mathbf{x}_{i_2}^2) = \int_{K_j} G(\boldsymbol{\mu};\mathbf{x}_{i_2}^2,\mathbf{y})\; G(\boldsymbol{\mu};\mathbf{y},\mathbf{x}_{1_1}^1) d\mathbf{y}. \tag{46}$$

For large problems (a large number of sources, receivers and model parameters), the computation of this matrix can be very expensive and can be achieved only thanks to high–performance computing. However, it can be easily evaluated in the case of homogeneous media (i.e. $\mu(\mathbf{y}) = \mu_0$)

$$G(\boldsymbol{\mu};\mathbf{x},\mathbf{y}) = \frac{e^{\mathbf{i}kr}}{4\pi r} \quad \text{ with } k = \sqrt{\mu_0} \text{ and } r = |\mathbf{x}-\mathbf{y}|. \tag{47}$$

Practically, this particular choice, can be seen as the initial guess for the acoustic media under study.

Remark: In practice, the number of rows of the Born matrix is much larger than the number of columns, i.e. $I \gg J$.

Let us prove that the Born Matrix defined in (46) can be approximated by a low-rank matrix under suitable assumptions on the location of receivers and sources. Most of the arguments that are present in this Section are rather similar to those developed in the multipole theory to solve the direct problem [5]. The

main ingredient to obtain a low-rank approximation of the Born matrix A is the so–called kernel-independent fast multipole method [2]. This method furnishes a tensorial approximation of the Green matrix under the following assumptions, see Figure 10:

i) The sources $\mathbf{x}^1_{i_1}$ are included in a 3D–dimensional box B_1

$$B_1 = \mathbf{x}^1_0 + [-d_1, d_1]^3 \tag{48}$$

with $\mathbf{x}^1_0 \in \mathbb{R}^3$, the center of the box and $d_1 > 0$ the size of the box.
ii) The receivers $\mathbf{x}^2_{i_2}$ are included in a 3D box B_2

$$B_2 = \mathbf{x}^2_0 + [-d_2, d_2]^3 \tag{49}$$

with $\mathbf{x}^2_0 \in \mathbb{R}^3$ the center of the box and d_2 the size of the box.
iii) The reflectors K_j are included in a 3D box B_3

$$B_3 = \mathbf{y}_0 + [-d_3, d_3]^3 \tag{50}$$

with $\mathbf{y}_0 \in \mathbb{R}^3$ being its center and d_3 — the size of the box.
iv) The diameters of these three boxes are smaller or equal to a wavelength

$$d_1 < \lambda, \quad d_2 < \lambda \quad \text{and} \quad d_3 < \lambda. \tag{51}$$

v) The distance D_1 between the boxes B_1 and B_3 and the distance D_2 between the boxes B_2 and B_3 are larger than some wavelengths.

$$D_1 \gg \lambda \quad \text{and} \quad D_2 \gg \lambda. \tag{52}$$

In the box B_ℓ, $1 \leq \ell \leq 3$, we can approximate the function $f_\ell : B_\ell \longrightarrow \mathbb{C}$ in the

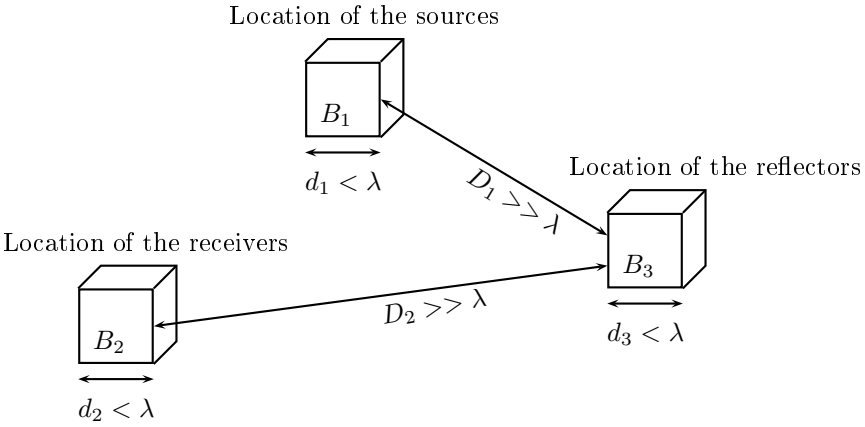

FIG. 10. Assumptions on the sources, receivers and reflectors

following way

$$f_\ell(\mathbf{x}) = \sum_{m=1}^{M} f_\ell(\mathbf{X}^\ell_m)\, \mathsf{p}^\ell_m(\mathbf{x}) + \varepsilon_\ell(\mathbf{x}) \qquad \forall f_\ell \in C^\infty(B_\ell), \tag{53}$$

where for $1 \leq m \leq M$ and $\ell \in [1,3]$, $\mathbf{X}^\ell_m \in \mathbb{R}^3$ are the interpolation points the $\mathsf{p}^\ell_m : B_\ell \longrightarrow \mathbb{R}$ are interpolating functions. The residual ε^ℓ_M is small when the

interpolating function is well chosen (see Appendix C for a possible choice) and the interpolated function is regular.

In the context of the Born matrix, these interpolation functions can be used to define a tensorial approximation of the Green function

$$(54)\quad \begin{cases} G_k(\mathbf{x},\mathbf{y}) \simeq \displaystyle\sum_{m=1}^{M}\sum_{n=1}^{M} G_k(\mathbf{X}^1_m,\mathbf{X}^3_n)\mathsf{p}^1_m(\mathbf{x})\mathsf{p}^3_n(\mathbf{y}), & \mathbf{x}\in B_1,\ \mathbf{y}\in B_3,\\ G_k(\mathbf{x},\mathbf{y}) \simeq \displaystyle\sum_{m=1}^{M}\sum_{n=1}^{M} G_k(\mathbf{X}^2_m,\mathbf{X}^3_n)\mathsf{p}^2_m(\mathbf{x})\mathsf{p}^3_n(\mathbf{y}), & \mathbf{x}\in B_2,\ \mathbf{y}\in B_3. \end{cases}$$

This brings about residual in the following approximation of the Born matrix

$$(55)\quad A_{i,j} \simeq \sum_{m_1=1}^{M}\sum_{m_2=1}^{M}\sum_{n_1=1}^{M}\sum_{n_2=1}^{M} G_k(\mathbf{X}^1_{m_1},\mathbf{Y}_{n_1})G_k(\mathbf{X}^2_{m_2},\mathbf{Y}_{n_2})$$
$$\mathsf{p}^1_{m_1}(\mathbf{x}^1_{i_1})\mathsf{p}^2_{m_2}(\mathbf{x}_{i_2})\int_{K_j}\mathsf{p}^3_{n_1}(\mathbf{y})\mathsf{p}^3_{n_2}(\mathbf{y})d\mathbf{y}.$$

Rearranging the latter, we deduce that the Born matrix A takes the form

$$(56)\qquad A = A^1\,A^2\,A^3,$$

with $A^1\in\mathbb{C}^{I\times M^2}$, $A^2\in\mathbb{C}^{M^2\times M^2}$ and $A^3\in\mathbb{C}^{M^2\times J}$

$$(57)\quad \begin{cases} A^1_{i,m} = \mathsf{p}^1_{m_1}(\mathbf{x}^1_{i_1})\mathsf{p}^2_{m_2}(\mathbf{x}_{i_2})\\ A^2_{m,n} = G_k(\mathbf{X}^1_{m_1},\mathbf{Y}_{n_1})G_k(\mathbf{X}^2_{m_2},\mathbf{Y}_{n_2})\\ A^3_{n,j} = \displaystyle\int_{K_j}\mathsf{p}^3_{n_1}(\mathbf{y})\mathsf{p}^3_{n_2}(\mathbf{y})d\mathbf{y} \end{cases}$$

where the integers m, $n\in[1,M^2]$ and $i\in[1,I]$ are related to the integers m_1, m_2, n_1, n_2, i_1 and i_2 by the relations

$$(58)\quad m=(m_1-1)\,M+m_2,\quad n=(n_1-1)\,M+n_2\quad\text{and}\quad i=i_1\,I_1+i_2+1$$

Equation (56) reveals that the Born matrix A can be approximated by a low-rank approximation when M^2 is much smaller than I and J. This is the case, when the number of receivers, sources and reflectors is very large.

We have proved that the Born matrix associated with a homogeneous acoustic medium admits a low-rank approximation under very restrictive assumptions on the location of the receivers, sources and reflectors. These results that use similar arguments to the fast multipole method can be extended. When these assumptions are not fulfilled, a low-rank approximation can also be obtained. It relies on elaborated arguments of the Fast Multipole Method. This will not be presented here due to its complexity.

For elastic media, it is also possible to show that the Born matrix associated with a homogeneous media admits a low-rank approximation. The reader can refer to [2] for the fast multipole method for elastic media.

Appendix C. The tensorial Chebyshev interpolation

We would like to briefly recall classical results about the Chebyshev tensorial interpolation of a function f in the box

$$B \;=\; \mathbf{x}_0 + [-d,d]^3. \tag{59}$$

We denote by C_P the Chebyshev polynomial of degree $P>0$ which is given by the formula

$$C_P(Z) \;=\; \cos(P\arccos(Z)), \quad Z\in[-1,1]. \tag{60}$$

For $1\leq p\leq P$, its zeros are denoted by $Z_p^P \;=\; \cos\big[(p-\frac{1}{2})\frac{\pi}{P}\big]$. To the zeros Z_p^P, that all belong to $[-1,1]$, we associate the Lagrangian interpolation polynomial

$$I_p^P(z) \;=\; \prod_{\substack{k=1\\k\neq p}}^{P} \frac{z-Z_k^P}{Z_p^P-Z_k^P} \qquad\qquad \text{for } p\in[1,P] \text{ and } z\in[-1,1]. \tag{61}$$

On the interval $[-1,1]$, any function u can be approximated by the formula

$$u(z) \;=\; \sum_{p=1}^{P} u(Z_p^P)\; I_p^P(z) \;+\; \varepsilon_P(z). \tag{62}$$

These interpolation polynomials are optimal in the sense that they minimize the L^∞–norm error

$$\|\varepsilon_P\|_{L^\infty([-1,1])} \;=\; \Big(\frac{2}{\pi}\log(P+1)+1\Big)\frac{(\pi/2)^P}{P!}\|u^{(P)}\|_{L^\infty([-1,1])}. \tag{63}$$

Thanks to this family of unidemensional interpolation functions, we define the tensorial interpolation functions $\mathsf{p}_m : B\longrightarrow\mathbb{C}$ on the box B

$$\mathsf{p}_m(\mathbf{x}_0+\mathbf{z}\,d) \;=\; I_{p_1}^P(z_1)\; I_{p_2}^P(z_2)\; I_{p_3}^P(z_3) \text{ with } \mathbf{z}=(z_1,z_2,z_3)\in[-1,1]^3. \tag{64}$$

and the interpolation points $\mathbf{X}_m\in B$

$$\mathbf{X}_m \;=\; \mathbf{x}_0 + d\,(Z_{p_1}^P, Z_{p_2}^P, Z_{p_3}^P) \tag{65}$$

where we have denoted by $m\in[1,M]$, with $M=P^3$, the integer defined by

$$m = p_3P^2+p_2P+p_1 \text{ with } p_1,p_2 \text{ and } p_3\in[1,P] \tag{66}$$

It follows that a function $f : B\longrightarrow\mathbb{C}$ can be approximated in the following way

$$f(\mathbf{x}) \;\simeq\; \sum_{m=1}^{M} \mathsf{p}_m(\mathbf{x})\; f(X_m) \tag{67}$$

References

[1] N. S. Bakhvalov, *Numerical Methods*, Nauka, 1973. in Russian. MR0362811

[2] S. Chaillat and M. Bonnet, *Recent advances on the fast multipole accelerated boundary element method for 3D time-harmonic elastodynamics*, Wave Motion, **50**:7 (2013), 1090–1104. MR3144050

[3] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, Society for Industrial and Applied Mathematics, Journal on Scientific and Statistical Computing, **11**:5 (1990), 873–912. MR1057146

[4] Z. Drmac and K. Veselic, *New fast and accurate Jacobi SVD algorithm* I, SIAM J. Matrix Anal. Appl., **35**:2 (2008), 1322–1342. Zbl 1221.65100

[5] A. Gumerov and R. Duraiswami, *Fast multipole methods for the Helmholtz equation in three dimensions*, Elsevier, Amsterdam, 2004.
[6] N. Halko, P. Martinsson, and J. A. Tropp, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review, **53**:2 (2011), 217–288. MR2806637
[7] I.Silvestrov, D.Neklyudov, C.Kostov, and V.Tcheverda. *Full-waveform inversion for macro velocity model reconstruction in look-ahead offset VSP: numerical SVD-based analysis*, Geophysical Prospecting, **61**:6 (2013), 1099–1113.
[8] G. Ming and S.C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM Journal on Scientific Computing, **17**:4 (1996), 848–869. Zbl 0858.65044
[9] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Acad. Press, New York, 1970. MR0273810
[10] P. P. M. De Rijk, *A one-sided Jacobi algorithm for computing the singular value decomposition on a vector computer*, SIAM J. Sci. Stat. Comp., **10** (1998), 359–371.
[11] S. Rjasanow, *Adaptive cross approximation of dense matrices*, In IABEM 2002, International Association for Boundary Element Methods, UT Austin, TX, USA, May 28-30, 2002.
[12] O. Schenk and K. Gartner, *On fast factorization pivoting methods for sparse symmetric indefinite systems*, ETNA. Electronic Transactions on Numerical Analysis [electronic only], **23** (2006), 158–179. MR2268558
[13] I. Silvestrov, D. Neklyudov, M.Puckett, and V. Tcheverda, *Resolution and stability analysis of offset VSP acquisition scenarios with applications to full-waveform inversion*, In SEG Technical Program Expanded Abstracts 2012, 2012.
[14] W.Menke, *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, Inc., New York, 1984.

Sergey Alexandrovich Solovyev
Institute of Petroleum Geology and Geophysics SB RAS,
pr. Koptyuga, 3,
630090, Novosibirsk, Russia
E-mail address: solovevsa@ipgg.sbras.ru

Sébastien Tordeux
Inria Bordeaux Sud-Ouest, Equipe-Projet Magique-3D IPRA-LMA
Université de Pau et des Pays de l'Adour
BP 1155, 64013 Pau Cedex, Université de Pau, France
E-mail address: sebastien.tordeux@gmail.com